Users' Guide

RIOT Model was designed to be a highly flexible and easily customized system. The user can easily specify:

1. the number of substances and grid points desired,
2. the size of the models,
3. the time step of the inner model,
4. the number of steps the inner model runs for before computing a single time step in the outer model,
5. the intervals at which information from the model is outputted
6. the interactions between different substances,
7. the initial concentration of each substance, the diffusion constants,
8. the outer model velocity field (constant throughout simulation), and,
9. the transfer of substances between the two models.

All of these can be controlled from the "`user_parameters.c`" file. However, before we go into the details of the control, we shall first explain the variables involved ("`variables.h`").

Throughout the simulation, all information is stored in two C data structures: `outer_field`, and `bacteria_field`. The `outer_field` holds information about the outer model, and likewise for the `bacteria_field`. In most of the source code, the `outer_field` is denoted by the variable `f`, and the `bacteria_field` is denoted by variable `b`. The concentration of species is stored in an array called `conc`, found in both f and b. f also contains an array of `b`'s, which basically mimic the idea that there is a representative cylinder found in each of the outer model. In the "`user_parameters.c`" file, functions that call on f deal with the outer model, while functions that call on b deal with the inner model.

The functions are arranged in sequence of user specification. We shall cover each of these functions accordingly.

User control functions ("user_parameters.c")

Function 1: initialize_outer_parameters (outer model)

Initializes parameters necessary for the running of the outer model.

Code variable	What it means
f->length_scale	Length unit which all outer quantities use. Specify in terms of metres.
f->time_scale	Time unit which all outer quantities use. Specify in terms of seconds.
f->mol_scale	Units of particle number which all outer quantities use. Specify in terms of mol.
f->dims[0]	Number of substances involved. This must be the same as for the inner model
f->dims[1]	Number of cubes in the x-direction
f->dims[2]	Number of cubes in the y-direction
f->dims[3]	Number of cubes in the z-direction
f->l[0]	Length of the cuboid in the x-direction (in specified length unit)
f->l[1]	Length of the cuboid in the y-direction (in specified length unit)
f->l[2]	Length of the cuboid in the z-direction (in specified length unit)
f->duration	Length of time over which the entire RIOT Model runs (in specified time unit)
f->outputInterval	Time interval at which RIOT Model outputs data (in specified time unit)
f->internal_steps_per_outer_steps	Number of inner model time steps before the outer model step runs.
f->mu	Newtonian fluid viscosity (not implemented)

Function 2: initialize_bacteria_parameters (inner model)

Initializes parameters necessary to run the inner model.

Code variable	What it means
b->length_scale	Length unit which all inner quantities use. Specify in terms of metres.
b->time_scale	Time unit which all inner quantities use. Specify in terms of seconds.
b->mol_scale	Units of particle number which all inner quantities use. Specify in terms of mol.
b->dims[0]	Number of substances involved. This must be the same as for the outer model
b->dims[1]	Number of rings
b->dims[2]	Number of rings required to stack up the vertical cylinder
b->l[0]	Radial length of the cylinder (in specified length unit)
b->l[1]	Vertical length of the cylinder (in specified length unit)
b->dt	Time step of the inner model (in specified time unit)

Function 3: initialize_outer_diffusion_constants

Initializes the diffusion constants of the various substances for the outer model

Code variable	What it means
f->D	1D array containing the diffusion constants of the various substances,

	in terms of the outer model units.

Function 4: initialize_bacteria_diffusion_constants

Initializes the diffusion constant for the various substances in the inner model.

Code variable	What it means
`b->D`	1D array containing the diffusion constants of the various substances, in terms of the inner model units.

Function 5: initialize_outer_flow

Initializes the fluid flow field for the outer model.

Code variable	What it means
`f->u`	x-direction fluid velocity (in terms of outer length and time units)
`f->v`	y-direction outer fluid velocity (in terms of outer length and time units)
`f->w`	z-direction outer fluid velocity (in terms of outer length and time units)

Note that the 3 fluid flow quantities have 3 indices. From left to right, they are: [x][y][z]

Function 6: initialize_outer_conc

Generates the starting concentration fields for the outer model.

Code variable	What it means
`f->conc`	Concentration for each substance at each point in terms of outer length and particle number units. Note that the indices are: [][x][y][z].

Function 7: initialize_outer_pop

Generates the starting concentration fields for the inner model.

Code variable	What it means
`f->pop`	Population density of bacteria in each outer point. Note that the indices are: [x][y][z].

Function 8: initialize_inner_conc

Generates the starting concentration fields for the outer model.

Code variable	What it means
`b->conc`	Concentration for each substance at each ring in terms of outer length and particle number units. Note that the indices are: [n][r][z].

Function 9: user_bacteria_interactions

Computes the rate of change of substances in the bacteria due to interactions. This basically handles the $R_i(\boldsymbol{c})$ term in the inner model diffusion-advection equation.

Code variable	What it means
`b->i_conc_dt`	Rate of change of substances due to interactions. Note that the indices are: [n][r][z].

Function 10: user_bacteria_wall_flux

Computes the exchange of substances between the inner and outer model. Currently set up to handle the exchange as free diffusion.

Code variable	What it means
`b->d_conc_dt`	Rate of change of substances due to diffusion processes. Note that the indices are: [n][r][z].

Compilation and running

This model was written in C with no other additional packages that needs installing. All that is needed is the `gcc` compiler. At a later date, the model will be serially parallelized using OpenMP. However, at this moment, to compile this model, simply open Terminal and go to the directory where the RIOT Model scripts are stored. For Mac and Linux OS, evoke:

```
>> cd /path/to/RIOT Model/script/directory
>> gcc main.c -lm -O3 -o user_specified_name.out
```

where `user_specified_name.out` can be replaced by whatever name the user wants, as long as the suffix is "`.out`" (Eg, `RIOT Model.out`)

For Windows users, using a Linux emulator of some kind (eg, Cygwin) or the new Windows 10 Bash shell, evoke the following:

```
>> cd /path/to/RIOT Model/script/directory
>> gcc main.c -lm -O3 -o user_specified_name.exe
```

To run RIOT Model on Mac/Linux after compiling, evoke:

```
>> ./user_specified_name.out
```

Tor for the case of Windows, evoke:

```
>> ./user_specified_name.exe
```

Note that the instructions specified here for Windows has only been tested on Cygwin.

Model outputs

The results of the model runs are stored at time intervals specified by "user_parameters.c". There are two types of files that result: inner model outputs and outer model outputs. These are text files (".txt"). Here, we will cover how to interpret those output files. The user can then employ Mathematica or MATLAB to plot the information contained in the output files.

The inner model outputs are denoted by "inner_TIME.csv", where TIME is a 9 digit number indicating the number of inner model time steps (size dt) elapsed at the time of the output. Each entry in the csv file is composed of 6 integers and a float. They are interpreted as:

(i, j, k, n, r, z, conc)

Quantity	Description
i	Outer model x-index
j	Outer model y-index
k	Outer model z-index
n	n-th substance of the inner model.
r	Inner model r-index
z	Inner model z-index
conc	Concentration of the n-th substance at position (r, z) in the inner model, which is located at (i,j,k) point of the outer model.

The outer model outputs are denoted by "outer_TIME.csv", where TIME is a 9 digit number indicating the number of inner model time steps (size dt) elapsed at the time of the output. Each entry in the csv file is composed of 4 integers and a float. They are interpreted as:

(i, j, k, n, conc)

Quantity	Description
i	Outer model x-index
j	Outer model y-index
k	Outer model z-index
n	n-th substance of the inner model.
conc	Concentration of the n-th at the (i,j,k) point of the outer model.

The CSV formats allow easy retrieval for most mainstream plotting programmes and routines. Simply subset the information according to the indexing system to derive the desired concentration arrays for plotting.